Heuristic Optimisation

Part 12: Constraints in genetic algorithms

Sándor Zoltán Németh

http://web.mat.bham.ac.uk/S.Z.Nemeth

s.nemeth@bham.ac.uk

University of Birmingham

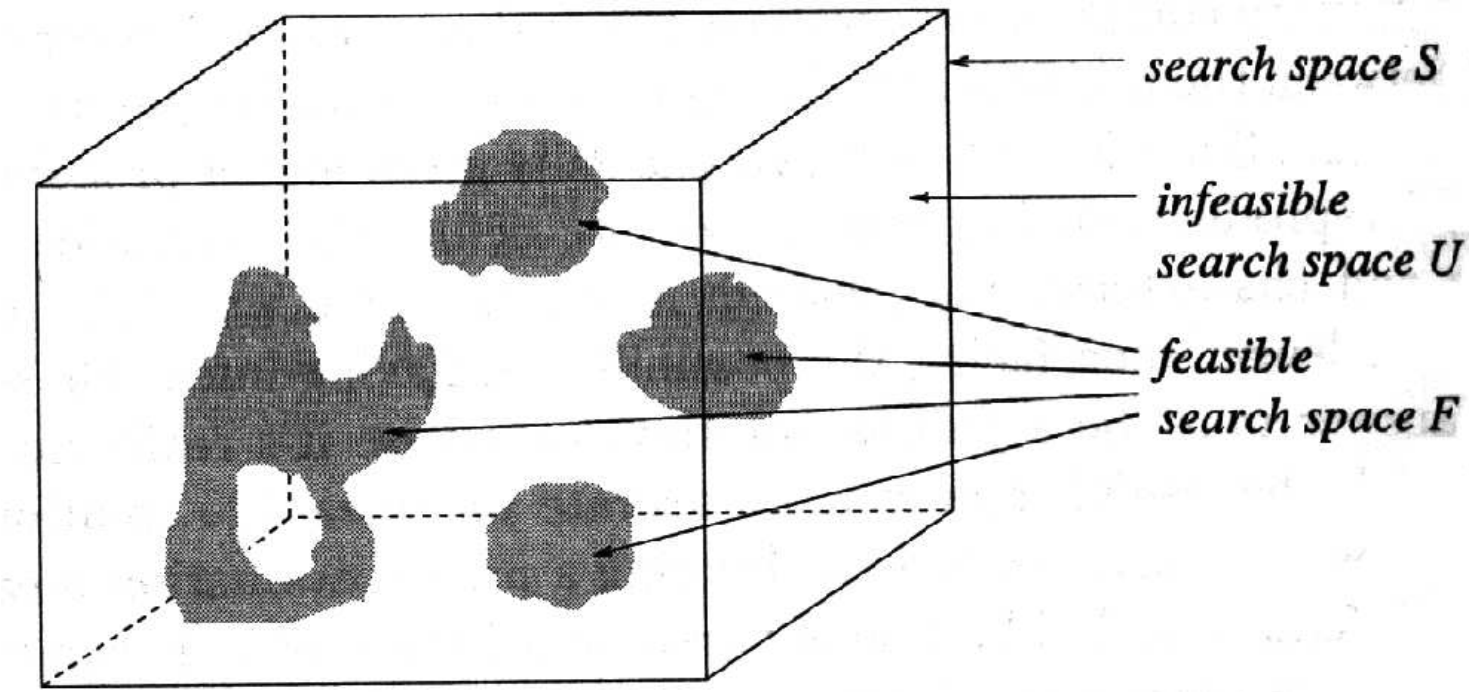
Overview

- Introduction
- Designing the evaluation function
- Rejecting infeasible individuals
- Repairing infeasible individuals
- Penalising infeasible individuals
- Using special representation, variation operators
- Using decoders

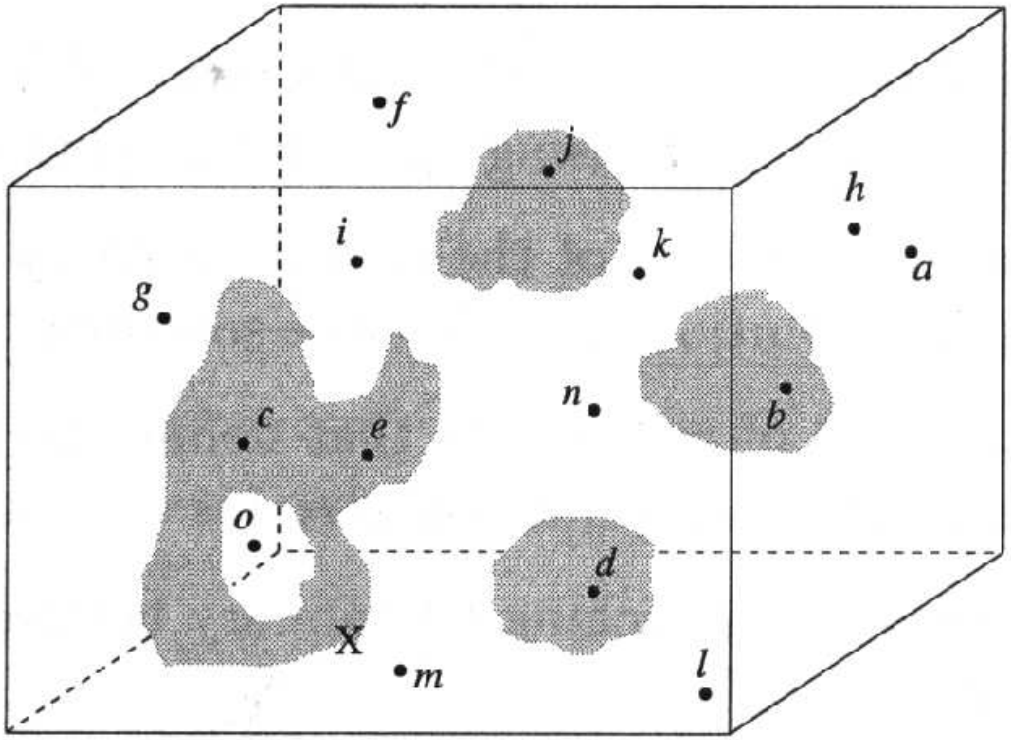
Introduction

In GAs better solutions have better chance of survival

The final solution must be feasible



Example



$(a, f, g, h, i, k, l, m, n, o)$ are infeasible

(b, c, d, e, j) are feasible

X is the global optimum

Questions

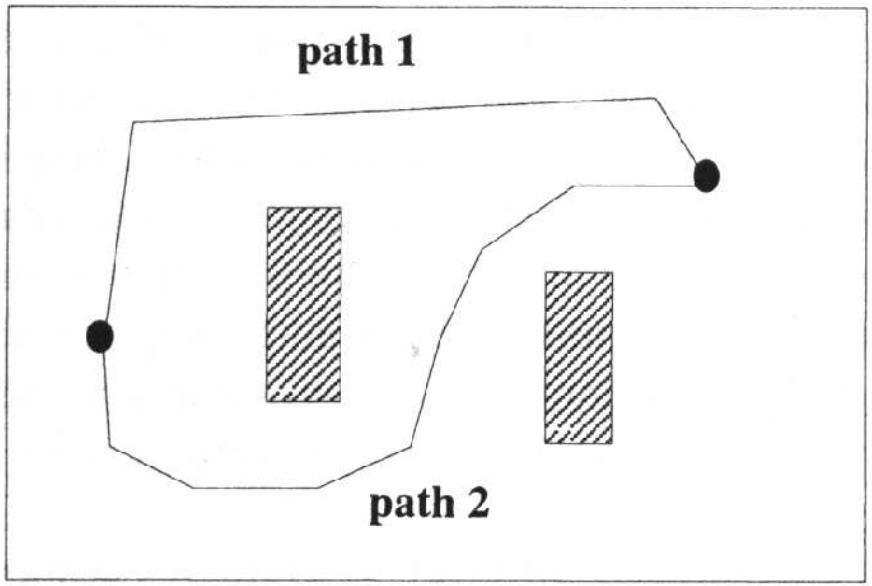
- How should we compare two feasible individuals (c and j)?
- How should we compare two infeasible individuals (a and n)?
- Should we assume that a feasible individual is always better than an infeasible one?
- Should we eliminate infeasible individuals from the population?
- Should we repair infeasible individuals?
- If we repair an infeasible individual, should we replace it by its repaired version?

Questions cont'd

- Should we penalise infeasible individuals?
- Should we start with a population of feasible solutions and maintain feasibility by using specialised operators?
- Should we use decoders?
- Should we consider individuals and constraints separately?
- Should we concentrate on searching the boundary between feasible and infeasible parts of the search space?

Evaluating feasible solutions

Which path is better for a robot?

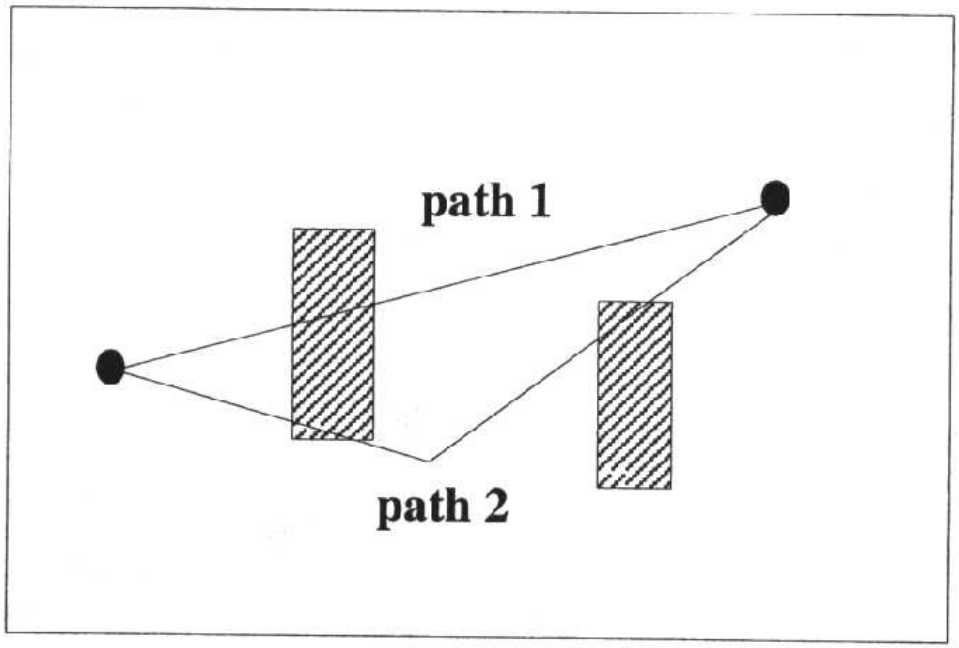


For the SAT problem:

$$F(x) = (\overline{x}_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_3)$$

$$eval_f(x) = x_1^2(x_2 - 1)^2(x_3 - 1)^2 + x_1^2 x_3^2$$

Evaluating infeasible solutions



$eval(x) = eval_f(x) \pm Q(x)$

$Q(x)$ can be a penalty or the cost of repairing x

Knapsack problem

Weight capacity 99 kilograms. Two infeasible solutions with the same total value

First: Total weight 100 kilograms

Second: Total weight 105 kilograms

Which is better?

Knapsack problem

Weight capacity 99 kilograms. Two infeasible solutions with the same total value

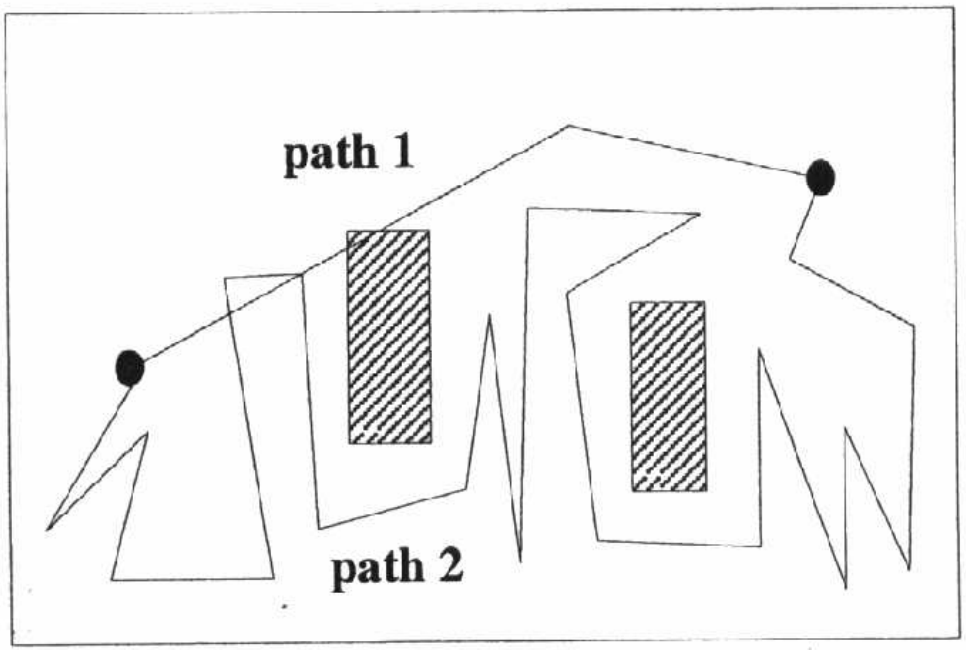
First: Total weight 100 kilograms

Second: Total weight 105 kilograms

Which is better?

First: Five items of 20 kilograms each
Second: An item of 6 kilograms of low profit

Feasible vs infeasible solutions



Which path is better for a robot?

Rejecting infeasible solutions

”Death penalty” is simple

Works well if the feasible part is convex and constitutes a reasonable part of the search space

Random sampling might give an initial population with infeasible solutions only

Variation operators might work better, if allowed to cross the infeasible region

Repairing infeasible solutions

The repaired version can be used for evaluation only or can replace the infeasible solution

The method is popular for combinatorial optimisation problems - TSP

Disadvantage: the method is problem dependent

Sometimes repairing an infeasible solution is just as hard as solving the problem

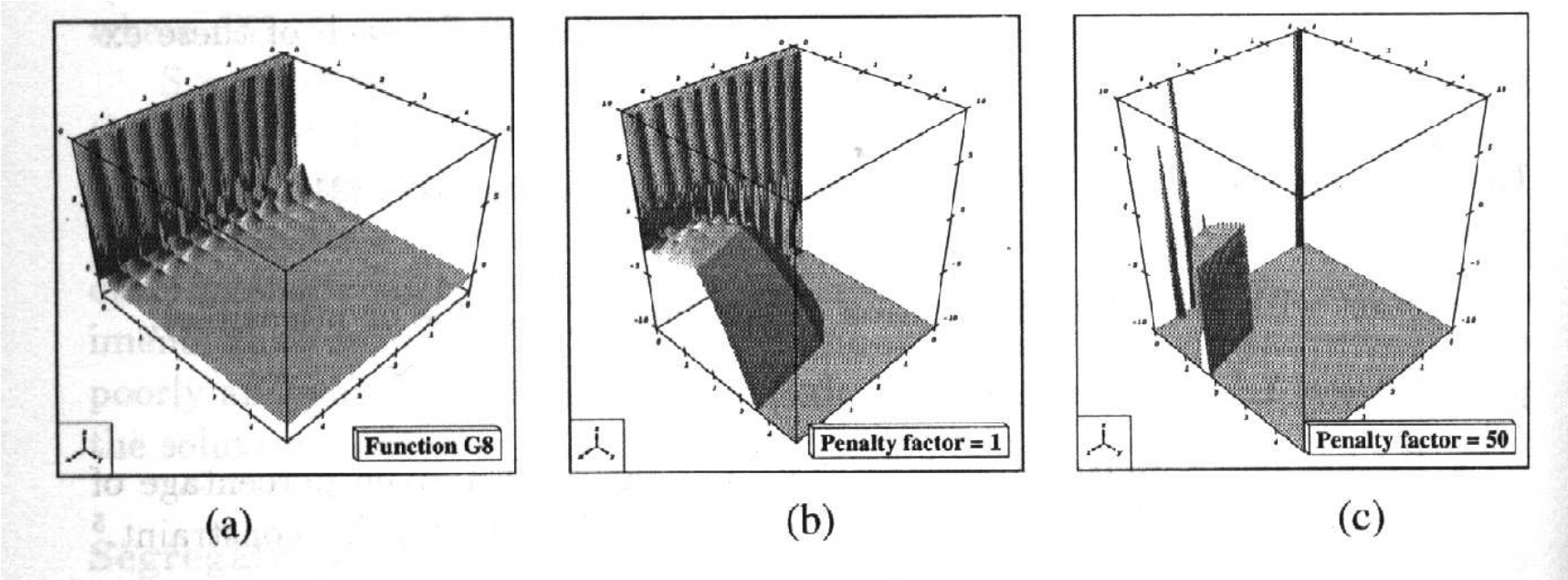
Penalising infeasible individuals

$$eval(x) = eval_f(x) \pm Q(x)$$

The penalty should depend on:

- the ratio between the size of the feasible region and the size of the search space
- the type of the evaluation function
- the number of variables
- the number of constraints
- the types of constraints
- the number of active constraints at the optimum

Different penalty factors



$$
\begin{cases}
\text{Maximize} & G_8(x) = \dfrac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)} \\
\text{subject to} & c_1(x) = x_1^2 - x_2 + 1 \le 0 \\
& c_2(x) = 1 - x_1 + (x_2-4)^2 \le 0 \\
& 0 \le x_1 \le 10 \\
& 0 \le x_2 \le 10
\end{cases}
$$

$$G_8^p(x) = G_8(x) - \alpha(c_1(x)^+ + c_2(x)^+)$$

Maintaining a feasible population

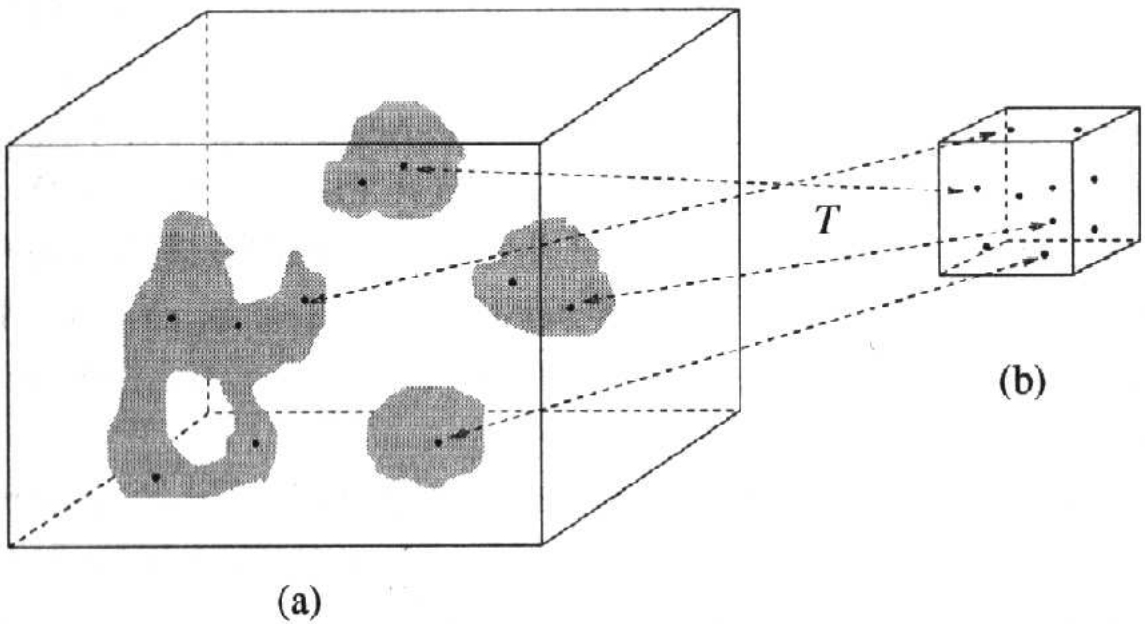
Use specialised representation and variation operators to keep within the feasible region

The variation operators transform feasible individuals into other feasible individuals

Works well for convex feasible region and linear constraints

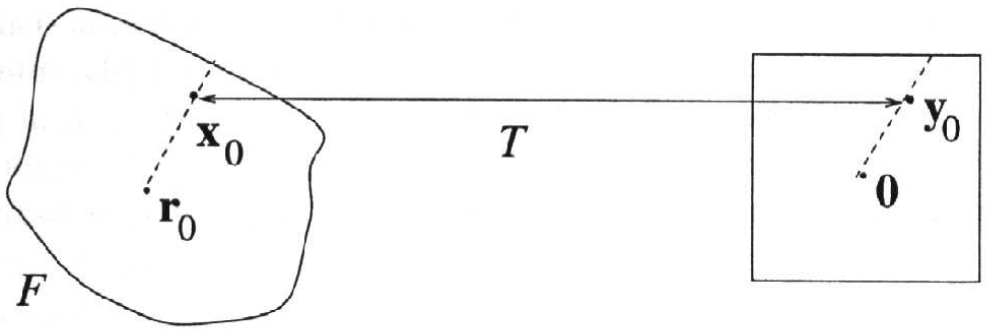
Using decoders

The representation is not for a solution, but for building a feasible solution



Example decoder

For continuous domains:



One-to-one mapping

Points that are close before the mapping are close after mapping

Decoder equation

$$\mathbf{y}_0 = (y_{0,1}, \ldots, y_{0,n}) \in [-1, 1]^n$$

$$\mathbf{y} = t\mathbf{y}_0, \; t \in [0, t_{\max}], \tag{1}$$

where

$$t_{\max} = \frac{1}{\max\{|y_{0,1}|, \ldots, |y_{0,n}|\}}$$

(1) = line segment from **0** to the boundary of the cube.

$$\mathbf{x}_0 = \mathbf{r}_0 + \tau\mathbf{y}_0, \tag{2}$$

where $\tau = \tau_{\max}/t_{\max}$.
(2) = decoder equation.

Summary

- In the majority of optimisation problems we encounter constraints
- There are many possible methods for handling constraints
- A careful analysis of the problem and the search space is needed before deciding which method to use

Recommended reading

Z. Michalewicz & D.B. Fogel
How to Solve It: Modern Heuristics

Chapter 9. Constraint-handling techniques